I'm creating an app on React and connecting to the database using mongodb. My app basically is an app which helps build connections with our community. I belong to the zoroastrian community. This app will specifically be only for our community members. People can register on our app and tell us their professional background, what work they do, what kind of business do they own, what kind of an entrepreneur they are. Now we as admins will hosts events which will facilitate getting to know more people from our community. For each event, we will reach out to those entrepreneurs within our community to give them business opportunities. So the event will be organised by an admin group with the help of such entrepreneurs and other zoroastrians can attend the attend just for the sake of having a fun time with fellow new zoroastrian friends.

Your app idea sounds like a fantastic way to strengthen community bonds while providing valuable business opportunities for members. Here's a general outline of how you can structure your app:

### 1. \*\*User Registration and Profiles\*\*

- \*\*User Registration:\*\* Allow users to sign up with basic information such as name, email, password, and optionally, a professional or business profile.

- \*\*User Profiles:\*\* Each user can create a profile detailing their professional background, business type, entrepreneurial activities, etc. This information can be used to connect them with relevant events and opportunities.

### 2. \*\*Admin Dashboard\*\*

- \*\*Event Management:\*\* Admins can create, manage, and publish events. You can include features like selecting entrepreneurs or businesses from the community to sponsor or organize events.

- \*\*User Management:\*\* Admins can view and manage registered users, including verifying their details and professional background.

- \*\*Notifications:\*\* Admins can send notifications to selected users, especially to entrepreneurs when a new event is being planned.

### 3. \*\*Event Creation and Participation\*\*

- \*\*Event Details:\*\* Events can have descriptions, dates, locations, and the businesses involved in organizing the event.

- \*\*RSVP System:\*\* Allow users to RSVP for events and see who else is attending.

- \*\*Feedback Mechanism:\*\* Post-event, users can provide feedback or reviews about the event and the businesses involved.

### 4. \*\*MongoDB Integration\*\*

- \*\*User Data:\*\* Store user profiles and their professional/business information in MongoDB.

- \*\*Event Data:\*\* Store all event-related information in MongoDB, including details about entrepreneurs and attendees.

- \*\*Analytics:\*\* Use MongoDB’s aggregation framework to analyze user data, event participation, and other metrics to help admins make data-driven decisions.

### 5. \*\*Front-End in React\*\*

- \*\*User Interface:\*\* Design a responsive and user-friendly interface for both users and admins. Consider using Material-UI or Bootstrap for styling.

- \*\*Forms:\*\* Implement forms for registration, profile creation, event creation, etc., with appropriate validation.

- \*\*Authentication:\*\* Use a library like `JWT` for handling user authentication and protecting routes.

### 6. \*\*Backend with Node.js and Express\*\*

- \*\*API Development:\*\* Create RESTful APIs using Express.js to interact with MongoDB, handle user authentication, manage events, and more.

- \*\*Role-Based Access Control:\*\* Implement role-based access control to distinguish between regular users and admins.